Chip-firing groups and fractal structures arising from graph-Laplacians

Jim Propp
UMass Lowell

December 4, 2007

slides available at
`http://jamespropp.org/chip.pdf`

Abstract: The chip-firing game has been invented at least twice: once as a pedagogical tool for teaching probability (the "stochastic abacus") and once as a testbed for ideas about the origins of complexity in the world ("self-organized criticality"). I'll convey the basic facts about chip-firing on directed graphs, focussing on the chip-firing group of a directed graph and its relationship to the Laplacian of the graph, and I'll show how some simple definitions give rise to complicated fractal structures whose nature has so far defied analysis.

0. Overview

Given a finite graph G with vertex set V and edge-set E, with multiple edges and self-loops allowed, and given a designated sink-vertex v^*, we create a group whose elements are certain non-negative integer-valued functions on $V \setminus v^*$.

This group is abstractly isomorphic to the cokernel of the reduced Laplacian of G (defined below), with a particular choice of representatives that appears to reflect some latent geometry in the graph G.

In particular, if G is a subset of a grid in the plane, the identity element in the chip-firing group of the graph G is an integer-valued function on $V \setminus v^*$ whose limiting behavior as the grid-size goes to zero appears to be fractal in many instances.

1. Defining the group

Let G be a connected graph with $d+1$ vertices.

For a running example to accompany the general definitions, we'll take G to be the cycle of length 3: $V = \{1, 2, 3\}$, $E = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$, $v^* = 3$.

The adjacency matrix of G is $A = (a_{i,j})$ where $a_{i,j}$ is the number of edges joining vertices i and j; the diagonal degree matrix is $D = (d_{i,j})$ where $d_{i,i}$ (also written as d_i) is the number of edges that contain vertex i, and $d_{i,j} = 0$ for $i \neq j$; and the Laplacian matrix is $L = D - A$.

Running example:

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}, \quad D = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix},$$

$$L = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix}$$

The Laplacian matrix is singular (each row-sum is zero).

The reduced Laplacian L^- is obtained from L by crossing out the row and column associated with the sink.

Running example:

$$L^- = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$$

The matrix L^- has full rank, so its cokernel $\mathrm{Z}^d/\mathrm{Z}^d L^-$ (where d = number of vertices in $V \setminus v^*$) is an abelian group whose order is the determinant of L^-. A fundamental parallelogram for the lattice spanned by the rows of L^- has vertices $(0,0)$, $(2,-1)$, $(-1,2)$, and $(1,1)$.

Coset representatives: $(0\ 0)$, $(0\ 1)$, $(1\ 0)$
Check: $(0\ 1) + (0\ 1) = (0\ 2) \sim (1\ 0)$.

Combinatorial model (Arthur Engel's chip-firing game; the Bak-Tang-Wiesenfeld sandpile model):

A non-negative vector $(a_1 \ldots a_d)$ in Z^d corresponds to a collection of indistinguishable chips, with a_i chips on the ith non-sink vertex of G. We call such a vector a *chip-configuration*.

One chip-configuration is a *successor* of another if it can be obtained from the other by subtracting a row of L^-; that is, remove d_i chips from the ith vertex and distribute them among the neighboring vertices, with $a_{i,j}$ chips going to vertex j. This is called firing vertex i.

Running example: $(2\ 3) \rightarrow (0\ 4)$.

We can only fire vertex i if vertex i is *active*, that is, if it has at least d_i chips on it. (In the case where i is a neighbor of the sink, it's sometimes helpful to picture things using the full Laplacian, with chips travelling from i to the sink and then becoming invisible.)

If we are not allowed to fire the sink, then it can be shown that the system eventually arrives at a configuration in which no site is active (i.e., no more firing is possible), which we call a *stable* configuration.

Running example: $(2\ 3) \to (0\ 4) \to (1\ 2) \to (2\ 0) \to (0\ 1)$

Strong convergence property: The stable configuration that arises from a given initial configuration is independent of the choices made about which active vertices to fire.

Running example: $(2\ 3) \to (3\ 1) \to (1\ 2) \to (2\ 0) \to (0\ 1)$

The ith toppling operator is the operation of adding a chip at vertex i and firing active vertices until a stable configuration results.

Abelian property: The toppling operators commute.

We call a chip-configuration accessible if it can be reached from any chip-configuration via chip-firing and addition of extra chips.

Running example: (1 1) is accessible, as is any configuration $(a\ b)$ with $a, b > 0$. Also, (0 1) and (1 0) are accessible.

Note that the stable configurations (0 0) and (1 1) are both equivalent modulo the reduced Laplacian, but only the latter is accessible.

A configuration that is both stable and accessible is called *recurrent*.

Theorem: Every equivalence class in Z^d mod L^- contains a unique representative that is recurrent.

These recurrent chip-configurations form a group under the composition law "add-and-stabilize".

Running example: Check this for (1 1), (0 1), and (1 0).

Note: The identity is (1 1), not (0 0).

Finding the identity element of a chip-firing group is non-trivial.

2. A simplified history

Arthur Engel: the stochastic abacus (e.g., expected number of steps a random walker takes to arrive at the sink)

Per Bak, Chao Tang, and Kurt Wiesenfeld: self-organized criticality

Deepak Dhar: the chip-firing group

3. Constructing the identity element

If we add the rows of the reduced Laplacian, we get a non-negative vector called the "burning vector" β; the ith component of β is the number of edges joining the ith vertex to the sink.

The burning vector is equivalent to the identity element, and we can use it to derive the identity element.

Adding the burning vector to a chip-configuration is like firing the sink.

[Show Stan Wagon's applet in the odd-length case]

[Show the burning vector]

We can use the burning vector β to compute the chip-firing group identity element several ways:

(a) start empty, add β, relax, add β, relax, ...

(b) start with 2β, relax, double, relax, ...

(c) start with 2β, fire all active sites and add β, fire all active sites and add β, ...; relax

[Demonstrate and explain!]

Running example, generalized: If G is a cycle with $d+1$ vertices, so that $V \setminus v^*$ is a path of d vertices, the identity element has 1's everywhere if d is even; it has 0 at the antipode of the sink and 1's everywhere else if d is odd.

4. Two dimensions

What happens if we take a simple non-stable configuration and stabilize it?

[Use Mike Creutz's **`xsand`** program to stabilize the all-4's configuration.]

[Use **`xsand`** to compute the chip-identity for various rectangles.]

Newsflash (Lionel Levine with technical assistance from David Wilson): It appears that when G is a mesh disk, the chip-firing identity is very close to being all-2's.

5. Directed graphs

The definition of the chip-firing group of a finite directed graph is similar.

Example: Let G be a directed cycle of length $d+1$ where

$a_{i,j} = 2$ if $j = i+1 \bmod d+1$,

$a_{i,j} = 1$ if $j = i-1 \bmod d+1$,

and

$a_{i,j} = 0$ otherwise.

Here's what the identity element looks like, for the first few values of n:

0
2 1
1 2 2
1 2 0 2
1 1 2 1 0
1 1 1 2 1 1
1 1 1 1 2 1 2
1 1 1 1 1 2 2 0
1 1 1 1 1 1 2 2 1
1 1 1 1 1 1 1 2 2 2
1 1 1 1 1 1 1 2 0 2 2
1 1 1 1 1 1 1 1 2 1 0 2
1 1 1 1 1 1 1 1 1 2 1 1 0
1 1 1 1 1 1 1 1 1 1 2 1 1 1
1 1 1 1 1 1 1 1 1 1 1 2 1 1 2

The rightmost digits are doing a funny sort of binary counting.

Another example: If we turn the n-by-n torus into a directed graph where each vertex has outdegree 2 so that the two successors of (x, y) are $(x + 1, y)$ and $(x, y + 1)$ mod 1, we get a picture with a fair bit of structure.

6. The stochastic abacus

See J. Laurie Snell, The Engel algorithm for absorbing Markov chains; `http://math.dartmouth.edu/~doyle/docs/engel/engel.ps`

7. Self-organized criticality

Scaling laws for sizes of avalanches

[Show Sergei Maslov's applet: `http://www.cmth.bnl.gov/~maslov/Sandpile.htm`

[Show the deviantart applet: `http://jayisgames.com/archives/2005/02/gridgame.php`